Security

UDP can be used, too

Security

UDP can be used, too

In a *UDP Flood* attack, the attacker(s) simply send very many UDP packets to the victim

Security

UDP can be used, too

In a *UDP Flood* attack, the attacker(s) simply send very many UDP packets to the victim

The victim OS is then overwhelmed by the need to read and process the packets and respond to them by returning an ICMP Destination Unreachable

Security

UDP can be used, too

In a *UDP Flood* attack, the attacker(s) simply send very many UDP packets to the victim

The victim OS is then overwhelmed by the need to read and process the packets and respond to them by returning an ICMP Destination Unreachable

A mitigation is to have a limit on the rate of ICMP error returns

Security

UDP can be used, too

In a *UDP Flood* attack, the attacker(s) simply send very many UDP packets to the victim

The victim OS is then overwhelmed by the need to read and process the packets and respond to them by returning an ICMP Destination Unreachable

A mitigation is to have a limit on the rate of ICMP error returns

Exercise Read about the *Low Orbit Ion Cannon* (LOIC)

Security

Recent botnets have used the *Internet of Things* (IoT), which is connected devices like security cameras, thermostats, doorbells, child monitors and so on

Security

Recent botnets have used the *Internet of Things* (IoT), which is connected devices like security cameras, thermostats, doorbells, child monitors and so on

They are often poorly secured, are still using default passwords, or are running old, vulnerable software

Security

Recent botnets have used the *Internet of Things* (IoT), which is connected devices like security cameras, thermostats, doorbells, child monitors and so on

They are often poorly secured, are still using default passwords, or are running old, vulnerable software

The *Mirai* botnet has been implicated in a DDOS attack of over 1TB/s

Security

Recent botnets have used the *Internet of Things* (IoT), which is connected devices like security cameras, thermostats, doorbells, child monitors and so on

They are often poorly secured, are still using default passwords, or are running old, vulnerable software

The *Mirai* botnet has been implicated in a DDOS attack of over 1TB/s

This was *DNS amplification* attack: the subverted devices make DNS lookup requests to servers with a reply address forged to that of the victim

Security

The DNS replies, which are much larger than the requests, are sent to the victim, causing a DOS

Security

The DNS replies, which are much larger than the requests, are sent to the victim, causing a DOS

And because the packets are coming from DNS servers, it is again hard to tell who initiated the attack

Security

The DNS replies, which are much larger than the requests, are sent to the victim, causing a DOS

And because the packets are coming from DNS servers, it is again hard to tell who initiated the attack

This is another flooding attack using UDP

Security

The DNS replies, which are much larger than the requests, are sent to the victim, causing a DOS

And because the packets are coming from DNS servers, it is again hard to tell who initiated the attack

This is another flooding attack using UDP

There are similar flooding attacks using other public services (such as time servers (NTP) and directory servers (LDAP)) exist

Security

“The ’S’ in IoT stands for security”

Anon

Security

Implementation attacks

Security

Implementation attacks

These exploit bugs in IP implementations

Security

Implementation attacks

These exploit bugs in IP implementations

Some hosts were vulnerable to oversized ping packets: the *Ping of Death*

Security

Implementation attacks

These exploit bugs in IP implementations

Some hosts were vulnerable to oversized ping packets: the *Ping of Death*

These were sent as forged fragments that, when reassembled, were much larger than expected and overflowed OS buffers in the receiving host

Security

Implementation attacks

These exploit bugs in IP implementations

Some hosts were vulnerable to oversized ping packets: the *Ping of Death*

These were sent as forged fragments that, when reassembled, were much larger than expected and overflowed OS buffers in the receiving host

The usual result is a crash: another denial of service

Security

To mitigate, we should just ignore ICMP packets that claim to be larger than the MTU: such packets are never generated naturally

Security

To mitigate, we should just ignore ICMP packets that claim to be larger than the MTU: such packets are never generated naturally

Or fix the reassembly code

Security

To mitigate, we should just ignore ICMP packets that claim to be larger than the MTU: such packets are never generated naturally

Or fix the reassembly code

Modern implementations check sizes are sensible before trying to reassemble fragments

Security

October 2020: Microsoft report a newly-discovered Ping of Death vulnerability in their IPv6 networking code

Security

October 2020: Microsoft report a newly-discovered Ping of Death vulnerability in their IPv6 networking code

Actually not a “ping” but an ICMP Router Advertisement, but it is easy to invoke and can crash (blue screen) any unpatched Windows

Security

October 2020: Microsoft report a newly-discovered Ping of Death vulnerability in their IPv6 networking code

Actually not a “ping” but an ICMP Router Advertisement, but it is easy to invoke and can crash (blue screen) any unpatched Windows

You might ask how there are still bugs like this in modern operating systems?

Security

Fragment bombs are like SYN floods in effect

Security

Fragment bombs are like SYN floods in effect

Too many fragments for packets that are never completed and so can't be reassembled

Security

Fragment bombs are like SYN floods in effect

Too many fragments for packets that are never completed and so can't be reassembled

This overflows fragment buffer space (where fragments are kept pending reassembly) and likely causes a denial of service, even a crash

Security

Fragment bombs are like SYN floods in effect

Too many fragments for packets that are never completed and so can't be reassembled

This overflows fragment buffer space (where fragments are kept pending reassembly) and likely causes a denial of service, even a crash

Again, implementations need to timeout and drop old fragments

Security

Many other exploits of implementation exist

Security

Many other exploits of implementation exist

Usually from the implementers making invalid assumptions about IP and assuming packets are all well-formed and correct

Security

Many other exploits of implementation exist

Usually from the implementers making invalid assumptions about IP and assuming packets are all well-formed and correct

- Jolt (aka sPING): fragmented ICMP packets
- Land attack. The source addresses on TCP SYNs are the same as the destination. The server tries to respond to itself
- Teardrop. Overlapping fragments cause problems on reassembly

Security

- New Teardrop (aka Bonk, Boink, Teardrop2). Overlapping fragments on a UDP packet reassemble to form a packet with an invalid header
- Zero length fragments. In some implementations these were stored but never used. Thus storage was exhausted
- And so on

Making a robust implementation is very hard!

Social Engineering Attacks

These are a pre-computer attack, formerly known as *confidence tricks*

Social Engineering Attacks

These are a pre-computer attack, formerly known as *confidence tricks*

If the machine is too hard to attack, attack the user instead

Social Engineering Attacks

These are a pre-computer attack, formerly known as *confidence tricks*

If the machine is too hard to attack, attack the user instead

Often this is much easier than a machine attack

Social Engineering Attacks

It could be as simple as phoning up a systems administrator and persuading them to give you a password to their machine

Social Engineering Attacks

It could be as simple as phoning up a systems administrator and persuading them to give you a password to their machine

- Pretend to be a supervisor and threaten to sack them if they do not comply
- Pretend to be a distraught user who has lost their password
- Anything else to unbalance them or get their sympathy

Social Engineering Attacks

It could be as simple as phoning up a systems administrator and persuading them to give you a password to their machine

- Pretend to be a supervisor and threaten to sack them if they do not comply
- Pretend to be a distraught user who has lost their password
- Anything else to unbalance them or get their sympathy

This is much easier than trying to crack a password by brute force

Social Engineering Attacks

Another attack is phishing

Social Engineering Attacks

Another attack is phishing

This is a form of impersonation to try and convince the user to hand over valuable information, such as credit card numbers

Social Engineering Attacks

A typical phishing attack is:

Social Engineering Attacks

A typical phishing attack is:

- the victim receives an email purporting to be from their bank asking them to update their personal details. The email provides a convenient WWW link

Social Engineering Attacks

A typical phishing attack is:

- the victim receives an email purporting to be from their bank asking them to update their personal details. The email provides a convenient WWW link
- The page looks plausibly like the bank's

Social Engineering Attacks

A typical phishing attack is:

- the victim receives an email purporting to be from their bank asking them to update their personal details. The email provides a convenient WWW link
- The page looks plausibly like the bank's
- The victim enters their details and sends them off

Social Engineering Attacks

A typical phishing attack is:

- the victim receives an email purporting to be from their bank asking them to update their personal details. The email provides a convenient WWW link
- The page looks plausibly like the bank's
- The victim enters their details and sends them off
- The email and Web page are fakes, so now the details are in the hands of criminals

Social Engineering Attacks

Similarly for many other attacks, such as The *419* or *Nigerian* fraud named after the South African police code used to identify this approach

Exercise Read about these

Firewalls

One way to reduce the impact of an attack is to prevent bad packets reaching the host in the first place

Firewalls

One way to reduce the impact of an attack is to prevent bad packets reaching the host in the first place

A *firewall* is a router/gateway that sits between a private network and the wider Internet and tries to protect the private network from attacks

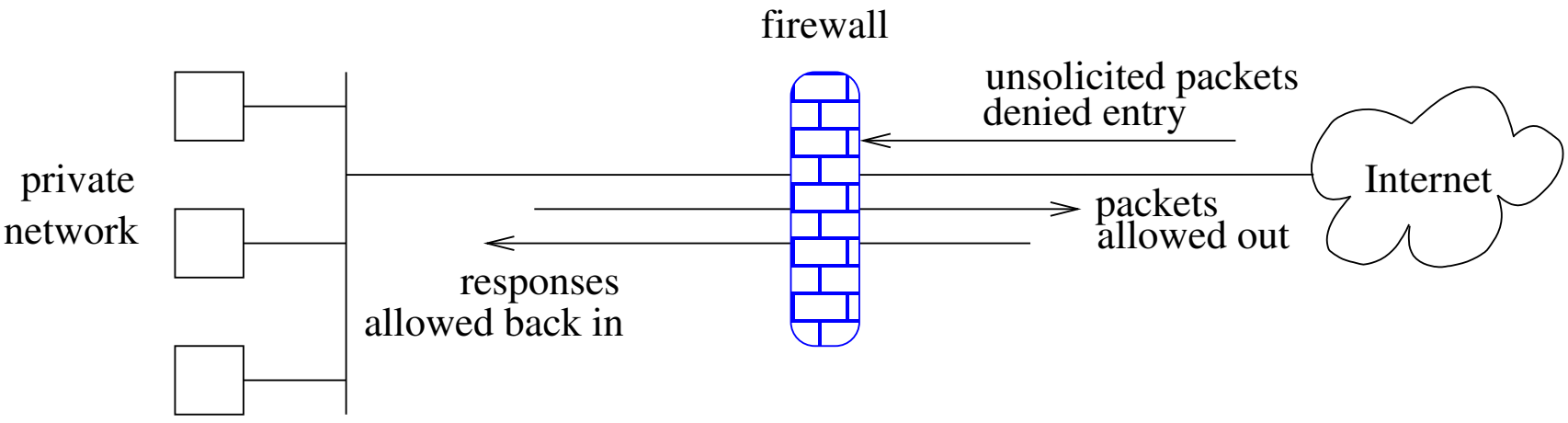
Firewalls

One way to reduce the impact of an attack is to prevent bad packets reaching the host in the first place

A *firewall* is a router/gateway that sits between a private network and the wider Internet and tries to protect the private network from attacks

It might be an ordinary router running firewall software, but specialised firewall hardware also exists

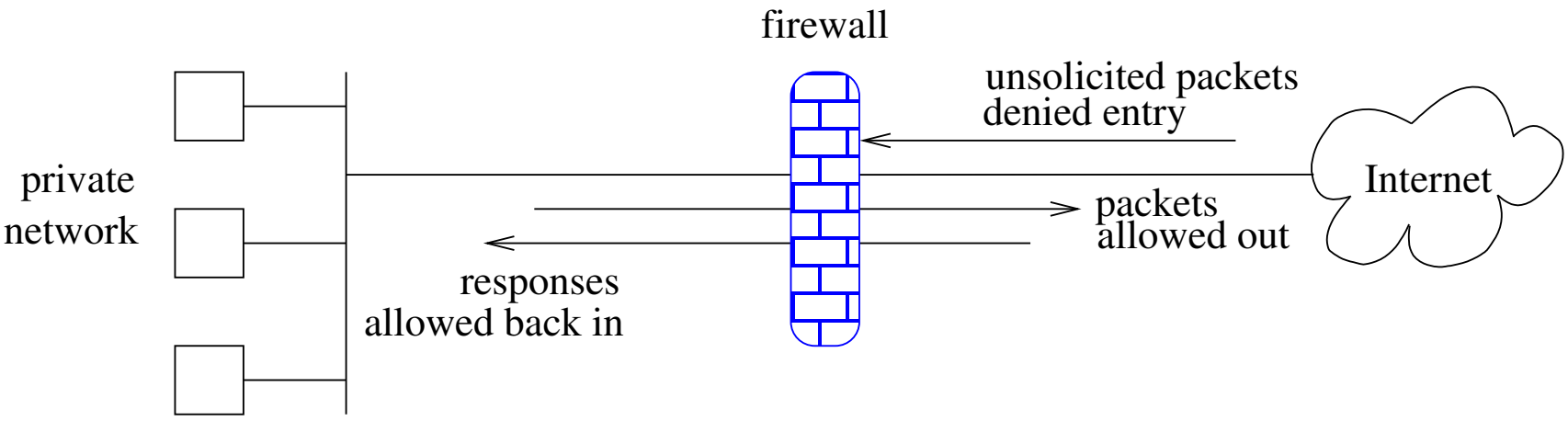
Firewalls



Firewall monitoring packets

The firewall inspects each packet as it enters and decides what to do with it. It might:

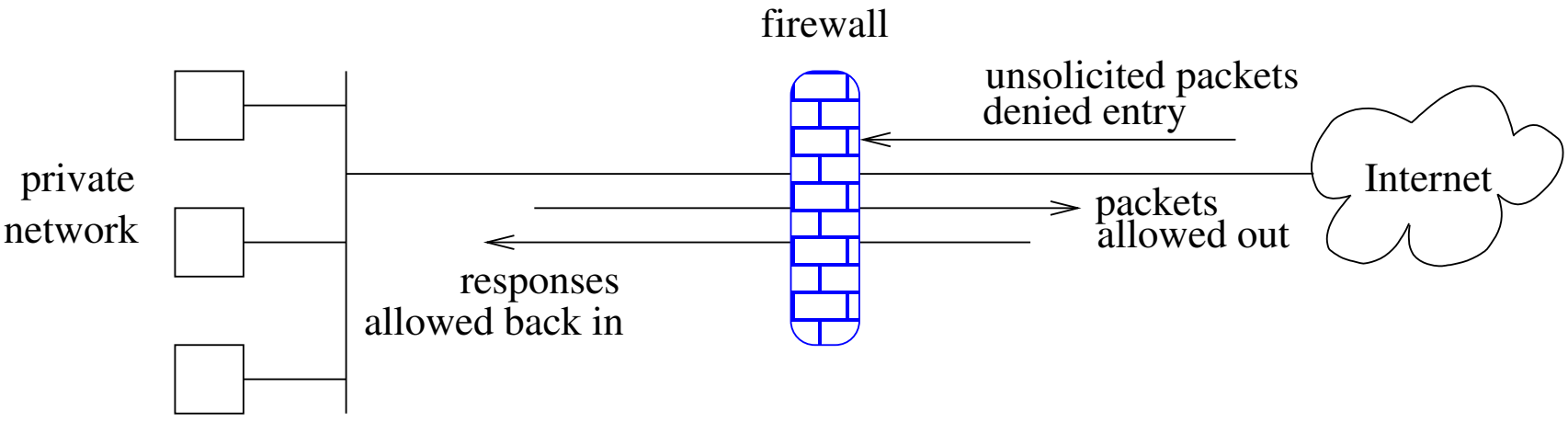
Firewalls



Firewall monitoring packets

Pass the packet through unchanged;

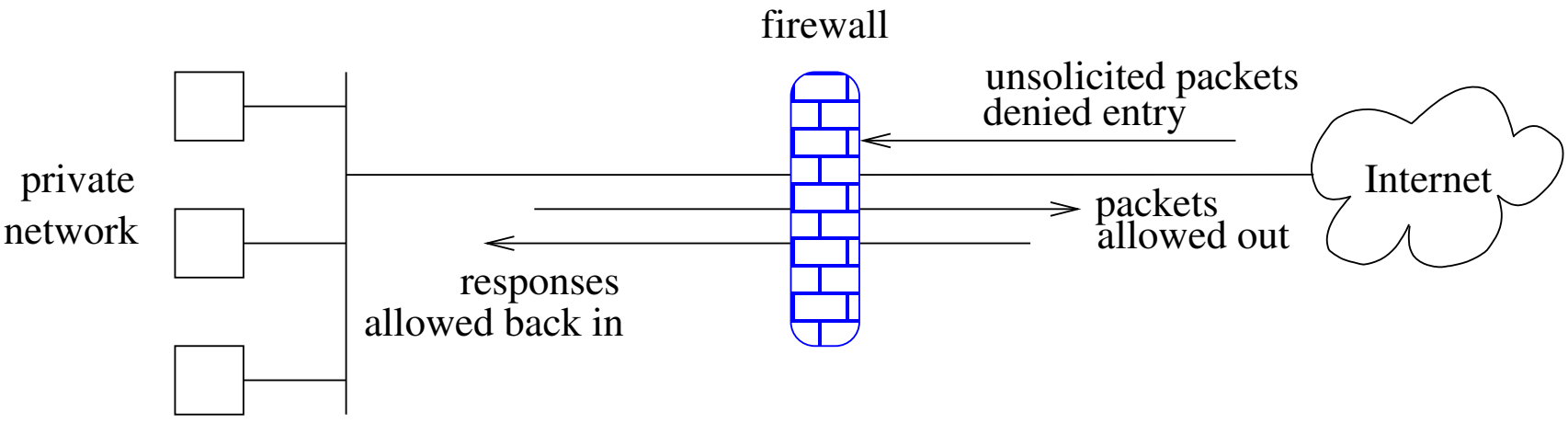
Firewalls



Firewall monitoring packets

Pass the packet through, but modified in some way, e.g., with the TOS bits changed or addresses changed with NAT;

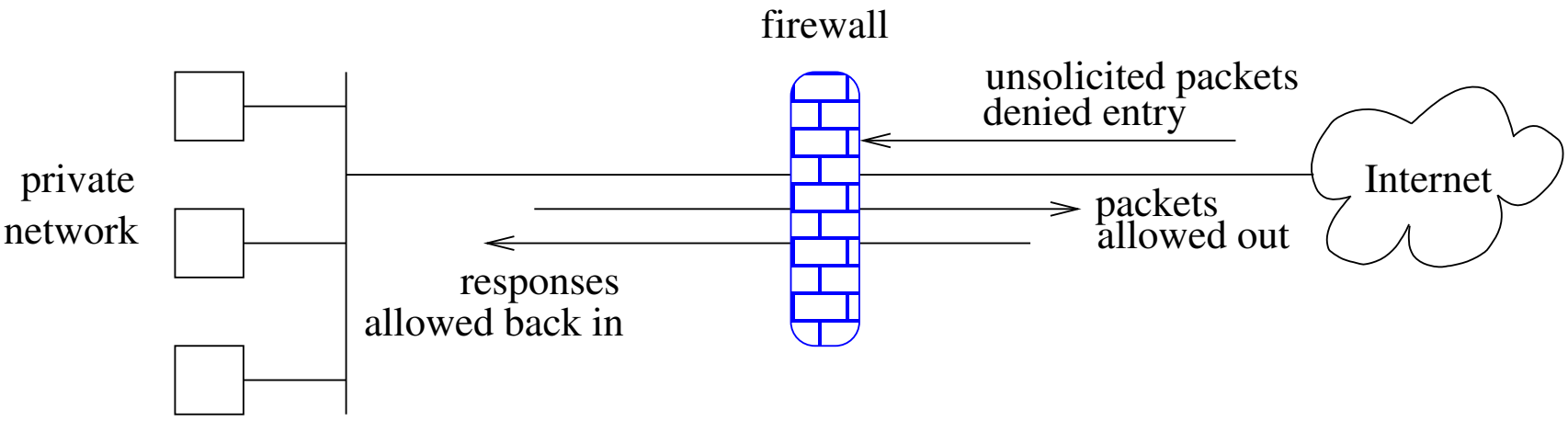
Firewalls



Firewall monitoring packets

Drop the packet and send an ICMP back, e.g., “port unreachable”;

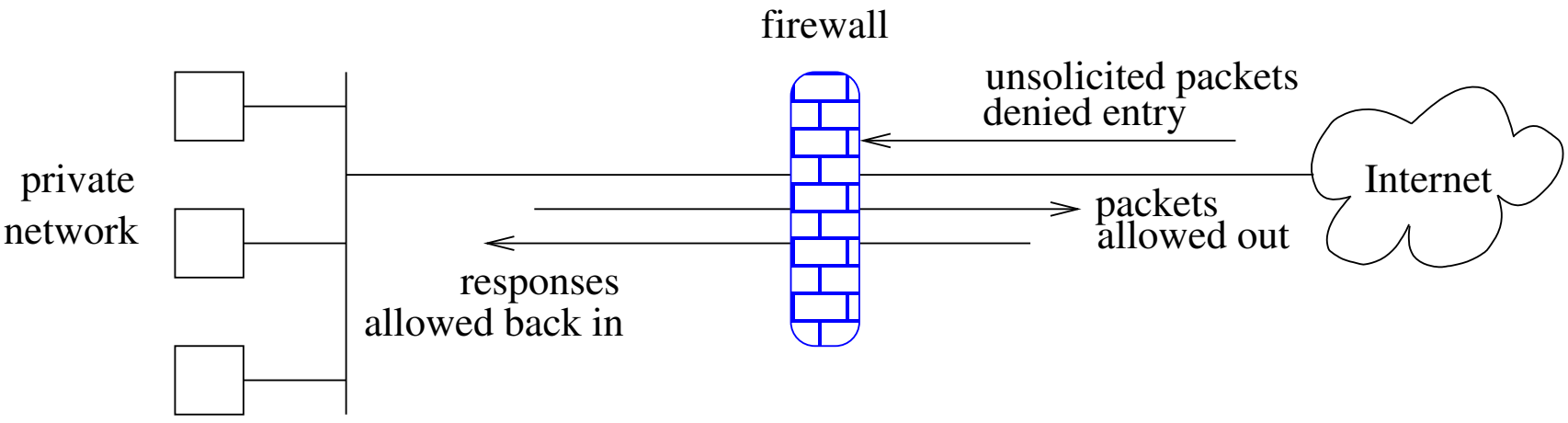
Firewalls



Firewall monitoring packets

Silently drop the packet;

Firewalls



Firewall monitoring packets

Or many other possibilities

Firewalls

Dropping silently is a good defence against probes from malicious sources looking for vulnerable services

Firewalls

Dropping silently is a good defence against probes from malicious sources looking for vulnerable services

The normal response to a packet sent to a closed UDP port is ICMP “port unreachable”; while TCP should send a RST

Firewalls

Dropping silently is a good defence against probes from malicious sources looking for vulnerable services

The normal response to a packet sent to a closed UDP port is ICMP “port unreachable”; while TCP should send a RST

But this has the side effect of telling the sender that this machine is up and running and worth probing further

Firewalls

Dropping silently is a good defence against probes from malicious sources looking for vulnerable services

The normal response to a packet sent to a closed UDP port is ICMP “port unreachable”; while TCP should send a RST

But this has the side effect of telling the sender that this machine is up and running and worth probing further

Silence can make the attacker believe there is no machine at that address at all

Firewalls

Dropping silently is a good defence against probes from malicious sources looking for vulnerable services

The normal response to a packet sent to a closed UDP port is ICMP “port unreachable”; while TCP should send a RST

But this has the side effect of telling the sender that this machine is up and running and worth probing further

Silence can make the attacker believe there is no machine at that address at all

Exercise Learn about scanning tools like `nmap`

Firewalls

Firewalling can be applied at any layer

Firewalls

Firewalling can be applied at any layer

The most common and useful are

Firewalls

Firewalling can be applied at any layer

The most common and useful are

- *packet filters* work in the data link, network and transport layers at the individual packet level, making decisions based on protocol (TCP or UDP, etc.), source and destination addresses, port numbers, TOS bits and so on

Firewalls

Firewalling can be applied at any layer

The most common and useful are

- *packet filters* work in the data link, network and transport layers at the individual packet level, making decisions based on protocol (TCP or UDP, etc.), source and destination addresses, port numbers, TOS bits and so on
- *application layer* firewalls work in the application layer and can use information that the applications use, e.g., HTTP filters can make decisions at the Web page level

Firewalls

There are also

Firewalls

There are also

- *application proxies* which also work in the application layer and act as an intermediate between the application and the server. They can also make use of application layer information

Firewalls

There are also

- *application proxies* which also work in the application layer and act as an intermediate between the application and the server. They can also make use of application layer information

A Web proxy for an institution might receive all HTTP requests from host within the organisation and choose to relay them onwards, or not, based the details of the HTTP request

Firewalls

Packet filters are fast, efficient and transparent to the application but do not have the discrimination available to application proxies

Firewalls

Packet filters are fast, efficient and transparent to the application but do not have the discrimination available to application proxies

Application layer filters are slower but more flexible

Firewalls

Packet filters are fast, efficient and transparent to the application but do not have the discrimination available to application proxies

Application layer filters are slower but more flexible

And proxies require some configuration in the application, e.g., setting up a Web browser to use a proxy

Firewalls

Packet filters are fast, efficient and transparent to the application but do not have the discrimination available to application proxies

Application layer filters are slower but more flexible

And proxies require some configuration in the application, e.g., setting up a Web browser to use a proxy

Of course, you can combine things: have a packet filter transparently rewrite packets to the Web to go via a proxy

Firewalls

Firewalls can protect services from attack from outside

Firewalls

Firewalls can protect services from attack from outside

E.g., not forwarding inward TCP packets that have destination port 21 will disallow external use of FTP into the private network

Firewalls

Firewalls can protect services from attack from outside

E.g., not forwarding inward TCP packets that have destination port 21 will disallow external use of FTP into the private network

This relieves some of the pressure of making all the FTP servers on the private network secure, but does not help against attacks from *inside* the firewall

Firewalls

Firewalls can protect services from attack from outside

E.g., not forwarding inward TCP packets that have destination port 21 will disallow external use of FTP into the private network

This relieves some of the pressure of making all the FTP servers on the private network secure, but does not help against attacks from *inside* the firewall

A safe default installation is to not forward anything inwards to effect maximum protection of the private network

Firewalls

Firewalls can protect services from attack from outside

E.g., not forwarding inward TCP packets that have destination port 21 will disallow external use of FTP into the private network

This relieves some of the pressure of making all the FTP servers on the private network secure, but does not help against attacks from *inside* the firewall

A safe default installation is to not forward anything inwards to effect maximum protection of the private network

This same protection is also a side-effect of NAT

Firewalls

Firewalls can protect services from attack from outside

E.g., not forwarding inward TCP packets that have destination port 21 will disallow external use of FTP into the private network

This relieves some of the pressure of making all the FTP servers on the private network secure, but does not help against attacks from *inside* the firewall

A safe default installation is to not forward anything inwards to effect maximum protection of the private network

This same protection is also a side-effect of NAT

Of course, NAT works nicely alongside firewalling

Firewalls

There can also be outward or *egress* filtering

Firewalls

There can also be outward or *egress* filtering

E.g., we can force the use of a HTTP proxy by internal hosts by blocking port 80

Firewalls

There can also be outward or *egress* filtering

E.g., we can force the use of a HTTP proxy by internal hosts by blocking port 80

More subtly, we could use NAT to rewrite connections to port 80 to the server running the proxy

Firewalls

There can also be outward or *egress* filtering

E.g., we can force the use of a HTTP proxy by internal hosts by blocking port 80

More subtly, we could use NAT to rewrite connections to port 80 to the server running the proxy

The proxy can then implement an application layer policy, e.g., disallowing access to certain web pages

Firewalls

There can also be outward or *egress* filtering

E.g., we can force the use of a HTTP proxy by internal hosts by blocking port 80

More subtly, we could use NAT to rewrite connections to port 80 to the server running the proxy

The proxy can then implement an application layer policy, e.g., disallowing access to certain web pages

Another, harsher, way of doing this is for the firewall to drop the packet and return a RST

Firewalls

There can also be outward or *egress* filtering

E.g., we can force the use of a HTTP proxy by internal hosts by blocking port 80

More subtly, we could use NAT to rewrite connections to port 80 to the server running the proxy

The proxy can then implement an application layer policy, e.g., disallowing access to certain web pages

Another, harsher, way of doing this is for the firewall to drop the packet and return a RST

Public wireless networks often block outward port 25 (SMTP) to prevent users sending spam

Firewalls

However, configuring a firewall is difficult and not to be taken lightly

Firewalls

However, configuring a firewall is difficult and not to be taken lightly

Capturing the many and varied requirements of a network is subtle and easy to get wrong

Firewalls

Exercise Some attacks get through the firewall by using a phishing attack to get a user to download and run some code. This code can then reach outwards through the firewall. Read about this

Exercise Some firewalls are configured to let in some traffic. For example, allowing an external connection to a security camera, so that you can remotely view your home. But if you can connect, so can others. Read about this

Exercise Some appliances, e.g., security cameras, connect outward to servers so that you can remotely view via the server. But if you can connect, so can others. Read about this

Security and Authentication in IP

The IP was not designed with security in mind

Security and Authentication in IP

The IP was not designed with security in mind

By default, the content of emails and web pages are readable as they travel to their destination

Security and Authentication in IP

The IP was not designed with security in mind

By default, the content of emails and web pages are readable as they travel to their destination

It is easy to write programs that trawl through millions of emails as they pass through a router

Security and Authentication in IP

The IP was not designed with security in mind

By default, the content of emails and web pages are readable as they travel to their destination

It is easy to write programs that trawl through millions of emails as they pass through a router

As a lot of sensitive and valuable data travels over the Internet these days we need to fix this

Security and Authentication in IP

The IP was not designed with security in mind

By default, the content of emails and web pages are readable as they travel to their destination

It is easy to write programs that trawl through millions of emails as they pass through a router

As a lot of sensitive and valuable data travels over the Internet these days we need to fix this

We need both security (encryption) and authentication

Security and Authentication in IP

We can apply these at any layer, e.g., in the IP model:

Security and Authentication in IP

We can apply these at any layer, e.g., in the IP model:

- Application. The application or the user can encrypt the data. For example, you might use PGP to encrypt an email before sending it. Or the application might have in-built encryption

Security and Authentication in IP

We can apply these at any layer, e.g., in the IP model:

- Application. The application or the user can encrypt the data. For example, you might use PGP to encrypt an email before sending it. Or the application might have in-built encryption
- Transport. SSL/TLS is described shortly. If trusting the user/application is too problematic we can get the transport layer to encrypt for us

Security and Authentication in IP

We can apply these at any layer, e.g., in the IP model:

- Application. The application or the user can encrypt the data. For example, you might use PGP to encrypt an email before sending it. Or the application might have in-built encryption
- Transport. SSL/TLS is described shortly. If trusting the user/application is too problematic we can get the transport layer to encrypt for us
- Network. At this layer we have IPSec, also described shortly

Security and Authentication in IP

We can apply these at any layer, e.g., in the IP model:

- Application. The application or the user can encrypt the data. For example, you might use PGP to encrypt an email before sending it. Or the application might have in-built encryption
- Transport. SSL/TLS is described shortly. If trusting the user/application is too problematic we can get the transport layer to encrypt for us
- Network. At this layer we have IPSec, also described shortly
- Data link. We can have encryption even in the data link layer. E.g., WPA is used to obscure wireless communications